

**ŽILINSKÁ UNIVERZITA V ŽILINE**

---

**AUTOREFERÁT  
DIZERTAČNEJ PRÁCE**

---

**Žilina, apríl, 2016**

**Ing., Boris Bučko**



**Žilinská univerzita v Žiline**  
**Fakulta riadenia a informatiky**

**Boris Bučko, Ing.**

Autoreferát dizertačnej práce

**Modelom riadená architektúra a ontológia**

na získanie akademického titulu „philosophiae doctor“ (v skratke **PhD.**)  
v študijnom programe doktorandského štúdia  
**aplikovaná informatika**

v študijnom odbore:  
**9.2.9 aplikovaná informatika**

Žilina, apríl 2016

**Dizertačná práca bola vypracovaná v externej forme doktorandského štúdia na  
Katedre informačných sietí, Fakulte riadenia a informatiky Žilinskej univerzity v  
Žiline**

**Predkladateľ:**                   **Ing., Boris Bučko**  
  **Katedra informačných sietí**  
  **Fakulta riadenia a informatiky**  
  **Žilinská univerzita v Žiline**

**Školiteľ:**                         **prof. Ing. Matilda Drozdová, PhD.**  
  **Katedra informačných sietí**  
  **Fakulta riadenia a informatiky**  
  **Žilinská univerzita v Žiline**

**Oponenti:**

**Autoreferát bol rozoslaný dňa: .....**

Obhajoba dizertačnej práce sa koná dňa ..... o ..... h. pred komisiou pre obhajobu dizertačnej práce schválenou odborovou komisiou v študijnom odbore **9.2.9 aplikovaná informatika, v študijnom programe aplikovaná informatika**, vymenovanou dekanom Fakulty riadenia a informatiky Žilinskej univerzity v Žiline dňa .....

**prof. Ing. Martin Klimo, PhD.**  
  predseda odborovej komisie  
  študijného programu **aplikovaná informatika**  
  v študijnom odbore **9.2.9 aplikovaná informatika**  
  Fakulta riadenia a informatiky  
  Žilinská univerzita  
  Univerzitná 8215/1  
  010 26 Žilina

## Úvod

Vývoj informačných systémov (IS) zaznamenal viac vývojových etáp, ktoré sú charakterizované tak z pohľadu ich využitia ako aj prístupu k riešeniu. Vznikali rôzne typy informačných systémov pre rôzne využitie. Čo však vo vývoji IS ešte stále chýba, je jednoznačné prepojenie požiadaviek používateľov do návrhu a následne implementácie systému. Metodiky vývoja IS začínajú vývoj prevažne špecifikáciou funkcionalít bez potrebných znalostí v kontexte implementačného prostredia. Z tohto dôvodu, nie sú vždy v dostatočnej miere akceptované také požiadavky implementačného prostredia, ktoré umožnia organizačné inovácie.

Proces vývoja informačných systémov je permanentnou úlohou softvérových architektov a vývojárov, ktorí sa snažia vyvíjať inovatívne riešenia a postupy. Na druhej strane sú to samotní používatelia, ktorých požiadavky sú výzvou k hľadaniu nových prístupov vývoja IS. Preto je v ostatnom období záujem hlavne o flexibilné a na údržbu nenáročné informačné systémy, ktoré dokážu splniť požiadavky používateľov v čo najkratšom možnom čase.

Väčšina informačných systémov nezlyháva z technických dôvodov, ale preto, že neriešia skutočné potreby zákazníkov. Nedostatočné upriamenie pozornosti na analytické etapy reprezentované v MDA (*Model Driven Architecture*) rovinou CIM (*Computation Independent Model*) iba odsúva skutočné problémy na neskôr, do implementácie a údržby systémov, kedy sú už nápravy chýb rádovo nákladnejšie.

## 1 Analýza súčasného stavu riešenej problematiky

Organizácie potrebujú využívať informačné technológie na podporu špecifických procesov. Vytvoriť informačnú podporu pre takéto špecifické procesy často krát nie je možné získať nakonfigurovaním existujúceho softvérového balíka. V takýchto situáciách je systém, či jeho súčasti nutné vytvoriť na mieru, priamo s ohľadom na požiadavky používateľov. Špecifické procesy vyžadujú logické a efektívne prepojenie dát, informácií a znalostí prostredníctvom implementácie riadenia znalostí resp. znalostného manažmentu.

Implementácia znalostného manažmentu z pohľadu modelovania vývoja IS je predovšetkým modelovanie roviny CIM. Ak všeobecne tvorba modelov pri vývoji IS zabezpečuje:

- systémový prístup,
- komplexnosť,
- prehľadnosť,
- spôsob vizuálneho vyjadrenia požiadaviek/kritérií IS,
- spôsob zachytenia životného cyklu vývoja IS,

potom pre implementáciu znalostného manažmentu je model CIM modelom najvyššej abstrakcie vo vývoji znalostných systémov.

Modelovanie v CIM rovine MDA a následná transformácia do roviny PIM dáva odpoveď na problémové otázky tvorby informačných systémov, ktoré sú nasledovné:

- a) Ako zachytiť funkcionality vytváraného informačného systému?
- b) Ako kritéria funkcionalít preniesť priamo do návrhu systému?
- c) Ako zabezpečiť adaptívnu zmenu kritérií funkcionalít?

Doterajšie riešenia modelovania CIM roviny v jazyku BPMN a následná transformácia do roviny PIM, uvedené vyššie, v mnohých prípadoch nie sú vhodné pre implementáciu znalostného manažmentu, pretože nedokážu zachytiť dostatočne sémantiku vykonávaných činností. Občas vznikajú nesprávne interpretácie jednotlivých BPMN diagramov, keď sa zamení význam plaveckých dráh s bazénmi (ak je súbor viacerých bazénov považovaný za plavecké dráhy a naopak), tak ako je to uvedené na [1]. Toto však súvisí skôr so samotným modelovaním ako so zachytením sémantiky, ktorá

je ale dôsledkom nesprávneho modelovania alebo úsudku následne aj chybne interpretovaná.

Modelovací jazyk BPMN (*Business Process Modelling Notation*) v modelom riadenej architektúre nie primárne koncipovaný na modelovanie znalostí, a nie je možné overiť sémantickú správnosť vytvorených modelov. Rovnako je problém zachovať sémantiku pri transformácií medzi jednotlivými úrovňami. Podporu modelovania znalostí pri vývoji znalostných IS by v tomto smere mohli poskytnúť ontológie. Preto sa ďalej v riešení zameriame na využitie ontológií pre vývoj znalostných IS.

Porovnaním charakteristík ODS (*Ontology Driven Systems*) a OES (*Ontology Enhanced Systems*) zistíme, že ontológia je väčším prínosom v ontologicky riadených systémoch. Pri ich vývoji je ontológia využitá od počiatočnej fázy a ontológiou je riešený komplexnejší problém v porovnaní s riešením OES, kde je ontológia iba doplnkom konvenčného postupu vývoja IS alebo doplnkom existujúceho IS. Aj keď vytváranie architektúry ODS vyžaduje oveľa viac práce ako návrh ontológie OES, správne navrhnutý architektonický rámec pre vývoj ODS môže eliminovať chyby spôsobené rôznymi typmi modelovacích jazykov v rôznych architektonických pohľadoch a transformáciami medzi nimi.

Rovina CIM v modelom riadenej architektúre vyjadruje model implementačného prostredia IS. V modernej informačnej spoločnosti, bez ohľadu na typ implementačného prostredia, vyjadruje rovina CIM modely informácií a dát, ich efektívne spracovanie, prepájanie a transformovanie na znalosti. Všetky činnosti všetkých podnikov a inštitúcií sú založené na zbere veľkého objemu informácií, ktoré sú ukladané ako dáta a zároveň sú potrebné ako znalosti pre prevádzku, riadenia a inovácie. Táto skutočnosť tak posúva ontológie do popredia v oblasti tvorby IS, predovšetkým znalostných IS.

Tým sa stáva využitie ontológie, ako nového prístupu pri vývoji informačných systémov, našou výskumnou výzvou.

## 2 Špecifikácia výskumného problému

Personalizáciu riešení IS je možné spojiť s pojmom *enterprise* architektúra, ktorý budeme ďalej používať ako pojem pre zdôraznenie prepojenia biznisu príslušnej organizácie alebo podniku a vyvíjaným informačným systémom. Všeobecnou paradigmou komplexného vývoja IS je v súčasnosti architektúra systému podľa normy ISO/IEC/IEEE 42010:2011, [2] a v nej špecifikovaný architektonický rámec, vytvorený z požadovaných pohľadov na vyvíjaný IS a vzájomných väzieb medzi pohľadmi. Pohľady vytvárajú parciálne architektúry, ktoré sú vytvárané v príslušných modelovacích jazykoch.

Informačné systémy pre podporu znalostného manažmentu sú zamerané na podporu znalostných techník na podporu rozhodovania, učenia a konania. Problematika znalostných systémov je široká a rieši rôzne problémy ich implementácie. Od celkového procesu vývoja, zladenie požiadaviek a potrieb používateľov, aplikácie rôznych znalostných metód, integráciu s konvenčnými technológiami, softvérové nástroje vývoja, rozhodovacie mechanizmy, interakciu používateľov, získavanie a reprezentáciu znalostí, jazyky a programovacie prostredie, techniky implementácie znalostí, systémové architektúry,....

Z pohľadu rovín modelom riadenej architektúry patrí znalostný manažment ako metóda do roviny CIM, ktorá v architektonických rámcoch vývoja IS predstavuje Biznis architektúru. Modelovanie CIM roviny v jazyku BPMN a dodržanie striktných pravidiel transformácie medzi modelmi CIM a PIM pre modelovanie princípov znalostného manažmentu nedokáže vždy správne vyjadriť sémantiku dát. Ak uvažujeme modelovanie podľa princípov znalostného manažmentu, je to modelovanie informácií, dát a znalostí. Informácie a znalosti patria v architektonickom rámci do biznis architektúry, dáta patria už do softvérovej architektúry, ktorá v MDA predstavuje rovinu PIM.

Ak ontológia dokáže riešiť vzťahy v CIM (informácie a znalosti) a transformácie CIM-PIM (informácie – dáta) mohla by byť vhodnejším nástrojom pre uplatnenie princípov v MDA, pretože nebude potrebné vytvárať transformačné vzťahy medzi CIM a PIM rovinou. Overenie nášho predpokladu je úlohou ďalšieho riešenia.



### 3 Cieľ dizertačnej práce

Na základe špecifikácie výskumného problému sme stanovili cieľ práce:

**Navrhnuť architektonický rámec riešenia informačného systému pre riadenie znalostí v organizácii použitím princípov modelom riadenej architektúry (MDA) a ontológií ako modelovacieho jazyka v rovinách CIM (*Computer Independent Model*) a PIM (*Platform Independent Model*).**

Dosiahnutie uvedeného cieľa je rozčlenené do 4 parciálnych cieľov:

1. Použitie ontológií pri vývoji IS vedie na novú disciplínu ontologické inžinierstvo. Ontologické inžinierstvo je pre oblasť IS najčastejšie spojené so znalostným inžinierstvom. Ak MDA je princíp vývoja IS, prvý cieľ riešenia je začleniť princípy MDA do postupov ontologického a znalostného inžinierstva.
2. Všeobecný vývoj IS podľa princípov normy ISO/IEC/IEEE 42010:2011 *Systems and Software Engineering* vedie na vytvorenie architektonického rámca, ktorý tvorí metodiku vývoja *software-intensive systems*. Vytvoriť architektonický rámec pre vývoj znalostných systémov s použitím ontológií s dôrazom na modelovanie roviny CIM a PIM je druhý parciálny cieľ.
3. Teoretické riešenie je potrebné overiť. Preto aplikovať vytvorený architektonický rámec pre špecifickú doménu s experimentálnym overením analýzy a návrhu IS je tretí parciálny cieľ.
4. Použitie ontológie ako modelovacieho nástroja v MDA môže odstrániť transformácie medzi rovinami CIM a PIM, ktoré sú potrebné pri použití grafických modelovacích jazykoch. Zdôvodniť a potvrdiť využitie ontológií v modelom riadenej architektúre oproti iným modelovacím jazykom je štvrtý parciálny cieľ.

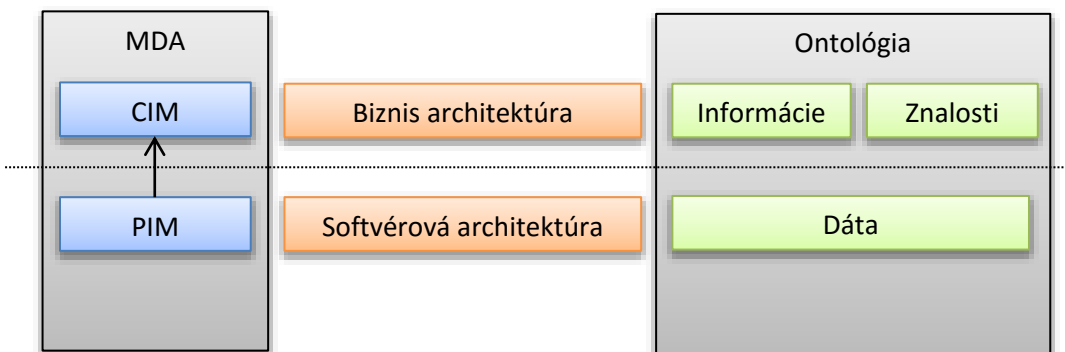
## 4 Riešenie

Nasledujúce podkapitoly prezentujú riešenie parciálnych cieľov tejto dizertačnej práce, ktoré vychádzajú z metód popísaných v predchádzajúcej kapitole.

### 4.1 MDA v ontologickom a znalostnom inžinierstve

Ontológia tvorí základ ontologického aj znalostného (ak je vybudované na ontológii) inžinierstva. Princípy MDA v súvislosti s ontológiou boli zhrnuté v kapitole 2, v podkapitole s názvom *Ontologie a MDA*.

Vrstva CIM – *Computer Independent Model* v MDA vyjadruje popis biznis architektúry a vrstva PIM – *Platform Independent Model* popis softvérovej architektúry. Naproti tomu ontológia zachytáva dáta informácie a znalosti. Určite existuje množstvo pohľadov, pomocou ktorých by sa dali vrstvy CIM/PIM prepojiť s ontológiou. Jeden takýto pohľad znázorňuje Obrázok 1. V ideálnom prípade by mohli byť vrstvy CIM a PIM reprezentované ontológiou, ktorá by mohla zjednodušiť proces transformácie, prípadne potrebu transformácie úplne odstrániť.



Obrázok 1 - Previazanie MDA s ontológiou

Spomenutým spôsobom je možné využiť princípy MDA v rámci ontologického a znalostného inžinierstva. Ontologie sú často súčasťou znalostných systémov, avšak tvorba znalostných systémov z architektonického hľadiska nie je triviálna záležitosť.

Nasledujúca podkapitola sa venuje tvorbe architektonického rámca pre vývoj znalostných systémov s použitím ontológií s dôrazom na modelovanie roviny CIM a PIM.

## 4.2 Návrh architektonického rámca pre znalostné systémy

Mať dobre navrhnutú architektúru je rozhodujúce pre úspech systému alebo podniku. Za účelom správneho návrhu systémovej architektúry je často využívaný architektonický rámec. Architektonický rámec možno podľa [3] charakterizovať ako opornú štruktúru/predpis, ktorá definuje architektonické artefakty, popisuje ako sú tieto artefakty prepojené medzi sebou a popisuje všeobecné definície toho, ako by tieto artefakty mohli vyzeráť.

Z pohľadu vývoja znalostných softvérových systémov možno tvrdiť, že ich princípy sú veľmi podobné so všeobecnými princípmi vývoja IS. Líšia sa po väčšine len v tom, že znalostné systémy musia riešiť nielen spracovanie informácie ale aj *manažment znalostí* – reprezentovanie, ukladanie a získavanie znalostí.

Pri návrhu architektonického rámca je treba zdôrazniť 3 elementy, ktoré sú súčasťou znalostného systému:

1. Dáta, informácie, znalosti.
2. Manažment znalostí.
3. Znalostné inžinierstvo.

Manažment znalostí pracuje s dátami, informáciami a znalosťami, ktoré s použitím znalostného inžinierstva (v tomto prípade je to asistencia znalostného inžiniera) spravuje a transformuje ich na znalosti, ktoré sú ukladané v znalostnej báze. Ďalej manažment znalostí prijíma dotazy, ktoré spracuje a následne v znalostnej báze vyhladá riešenie daného dotazu a poskytne odpoveď dotazovateľovi (napríklad používateľovi IS).

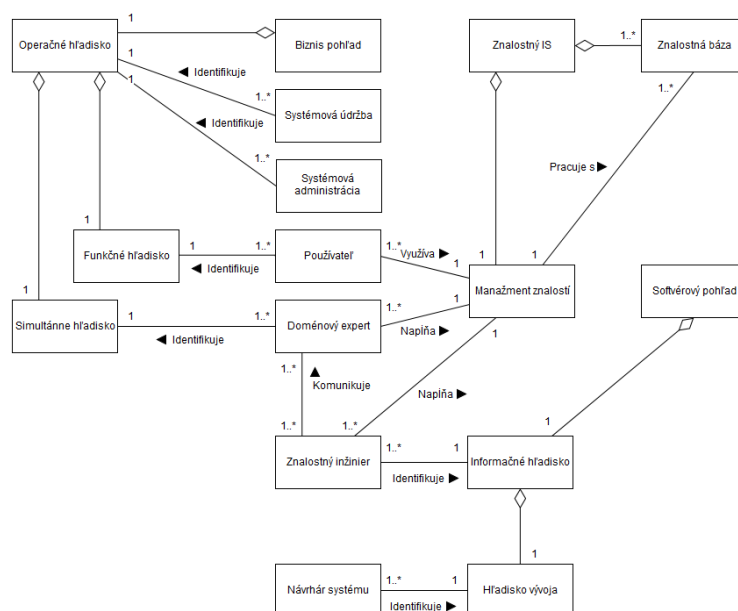
Na základe uvedených informácií o znalostných systémoch ale aj o zainteresovaných stranách v znalostných systémoch, vychádzajúc z rozdelenia hľadísk podľa [4] a ich následného zjednotenia do nadradených pohľadov je možné architektonický rámec znalostných systémov popísať nasledovne:

**Biznis pohľad** v sebe zahŕňa zainteresované strany – *používateľ* a *doménový expert*, ktoré priamo súvisia s funkcionálnym, simultánnym a operačným hľadiskom, pretože používateľ definuje a neskôr po nasadení systému využíva funkčné prvky

systemu (*funkčné hľadisko*) a takisto popisuje ako bude systém ovládaný (*operačné hľadisko*). To isté platí aj pre doménového experta, ktorý sa na systém pozerá z pohľadu znalostí, ktoré budú premietnuté do systému a navyše na základe jeho znalostí dokáže popísať súbežnosť systému (*simultánne hľadisko*). Ďalej biznis pohľad v rámci *operačného hľadiska* zahŕňa *systemovú údržbu* a *systemovú administráciu*.

**Softvérový pohľad** je tvorený *vývojármi systému* a *znalostným inžinierom*. Vývojári systému riešia *informačné hľadisko* a *hľadisko vývoja*. Znalostný inžinier v *informačnom hľadisku* integruje hľadisko doménového experta v rámci systému.

**Hardvérový pohľad** obsahuje *hľadisko nasadenia*, ktoré ako už bolo spomínané popisuje prostredie, do ktorého bude systém nasadený. V rámci hardvérového pohľadu bude vytvorené aj prostredie pre tvorbu a ukladanie znalostnej bázy. Ako už bolo spomínané vyššie hardvérový pohľad nie je v tejto práci bližšie špecifikovaný, a preto nie je zahrnutý ani v grafickom návrhu architektonického rámca pre znalostné systémy, ktorý ilustruje Obrázok 1.



**Obrázok 1 - Grafický návrh architektonického rámca pre znalostné systémy**

Obrázok je znázornený formou konceptuálneho modelu kontextu systému [5] podľa ISO/IEC/IEEE 42010:2011.

Vrstva CIM *Computer Independent Model* je vyjadrená *Biznis pohľadom*. Vrstva PIM – *Platform Independent Model* je reprezentovaná *Softvérovým pohľadom* (ale iba medziach platformovej nezávislosti). Ontológie sú zakomponované v rámci znalostnej bázy resp. v rámci znalostného systému. Týmto spôsobom navrhnutý architektonický zachováva dôraz na modelovanie roviny CIM a PIM a zároveň umožňuje využiť ontológie.

### **4.3 Využitie vytvoreného architektonického rámca**

Teoretický návrh architektonického rámca je potrebné prakticky overiť. Keďže sa jedná o architektonický rámec, ktorý je určený pre znalostné systémy, je vhodné zvoliť taký problém, ktorého vyriešenie požaduje využitie znalostí.

Overenie vytvoreného architektonického rámca bude demonštrované pri návrhu systému pre podporu plánovania procesov priemyselnej výroby.

Vhodné plánovanie výrobných procesov dokáže zabezpečiť nielen plynulú výrobu a dodávku tovarov, ale aj efektívne znížiť náklady, zrýchliť a optimalizovať proces celej výroby. Ambícia minimalizovať chyby v oblasti plánovania donútila priemysel automatizovať (resp. poloautomatizovať) aj samotný návrh a plánovanie výrobných procesov.

Zainteresované strany v systéme pre podporu plánovania priemyselnej výroby možno rozdeliť do niekoľkých skupín:

- Používatelia systému.
- Znalostní inžinieri.
- Doménoví experti.
- Vývojári systému.
- Systémoví administrátori.
- Dodávatelia systému.
- Systémová údržba.

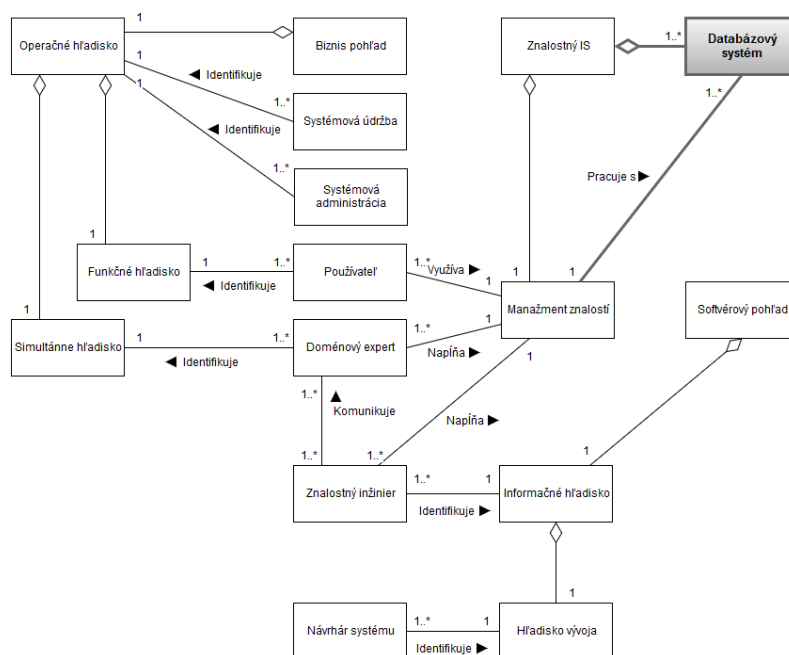
Keďže sa jedná o znalostný systém, v rámci návrhu bude potrebné upriamiť pozornosť na prepojenie systému so znalosťami, ktoré budú použité pri podpore

plánovania priemyselnej výroby. V informačných systémoch sú dáta, informácie a znalosti uchovávané napríklad v textovej podobe, v databázových systémoch alebo v ontológiách.

Uchovávanie znalostí v samotnom systéme pre podporu plánovania priemyselnej výroby sa môže líšiť prípad od prípadu a závisí od konkrétneho návrhu. V rámci overenia vytvoreného architektonického rámca autor zohľadňuje 2 spôsoby uchovávania znalostí:

- Databázový systém
- Ontológia

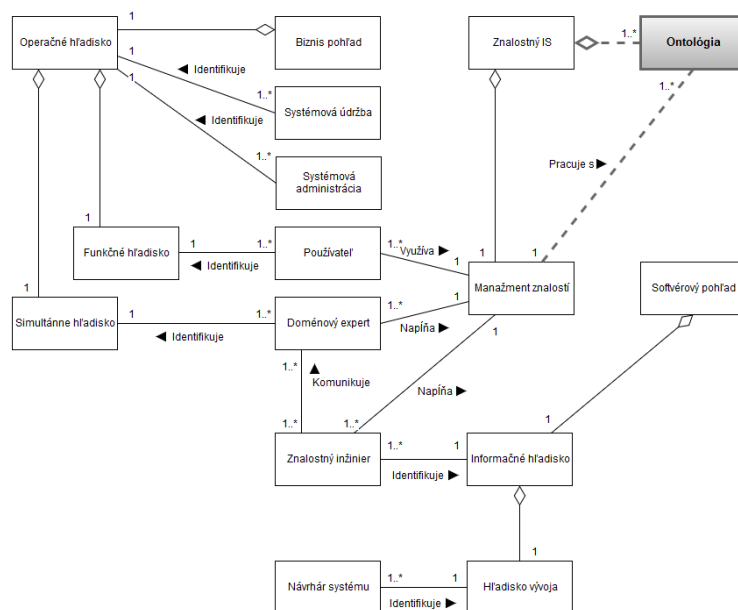
Pri použití databázového systému pre ukladanie znalostí by mohol architektonický rámec znalostných systémov vyzerat' nasledovne:



Obrázok 2 - Grafický návrh architektonického rámca pre znalostné systémy využívajúce databázový systém

Ako je možné vidieť, znalostná báza bola nahradená databázovým systémom. Výsledná podoba architektonického rámca sa rôzne prípady môže líšiť.

Pri využití ontológie je znalostná báza v návrhu architektonického rámca nahradená ontológiou, ktorá je vyňatá zo znalostného systému, aby mohla komunikovať aj s inými IS v rámci organizácie, tak ako je to znázornené na nasledujúcom obrázku.



**Obrázok 3 - Grafický návrh architektonického rámca pre znalostné systémy využívajúce ontológiu**

Vyňatie ontológie mimo znalostného systému je iba jeden zo spôsobov ako vnímať túto problematiku zo širšej perspektívy ale ontológia môže byť aj priamo integrovaná v znalostnom systéme. To či sa bude jednať o *ontologicky riadený systém* alebo o *systém rozšírený o ontológiu* závisí od konkrétneho návrhu.

Výsledným riešením experimentálneho overenia je desktopová aplikácia systému pre podporu plánovania procesov priemyselnej výroby vytvorená v platforme JavaFX. Analýza a návrh riešenia vychádza priamo z navrhnutého architektonického rámca, ktorého cieľom je sprehľadniť a uľahčiť etapu návrhu znalostného informačného systému.

Aplikácia poskytuje používateľovi podporu počas plánovania výrobných procesov na základe konfigurácie sledovaných parametrov a možných scenárov konkrétneho výrobného procesu. Medzi sledované parametre patria:

- Výroba konkrétneho druhu produktu alebo jeho časti,
- Počet pracovísk,
- Počet pracovníkov,
- Počet robotov,
- Časové presuny medzi pracoviskami
- Výška nákladov na vykonanie jednotlivých procesných operácií,

- Časová náročnosť jednotlivých procesných operácií,
- Časové verzus nákladové preferencie a ich úspora,
- Množstvo vyrobených kusov vzhľadom k nákladom a k spotrebovanému času.

Využitie aplikácie je ilustrované na príklade procesu výroby priemyselnej vidlice.



Obrázok 4 - Priemyselná vidlica - prebraté z [6]

Vidlica sa skladá z niekoľkých komponentov: *skrutka s nulovým kolíkom, skrutka s fázovým kolíkom, telo krytu, hlava vidlice, nálepka.*

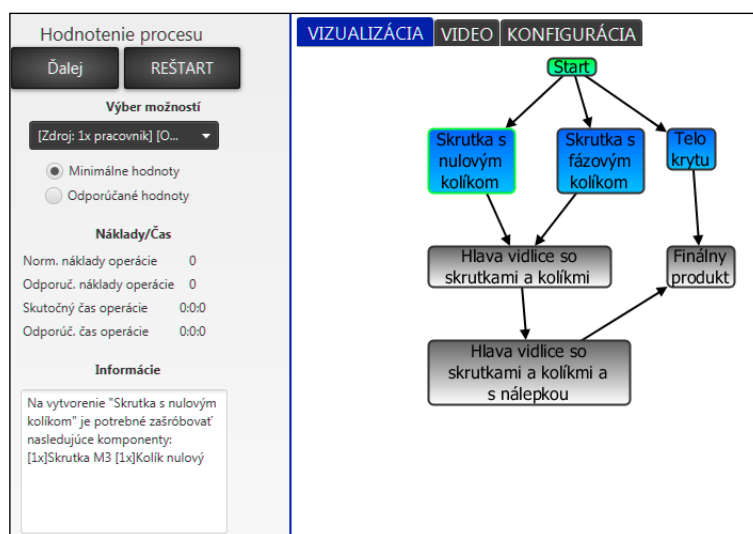
Zmontovanie priemyselnej vidlice podlieha výrobnému procesu, kedy je v danom okamihu možné montovať spolu iba niektoré komponenty. Takýmto spôsobom vznikajú medziprodukty (*hlava vidlice so skrutkami a kolíkmi, hlava vidlice so skrutkami a kolíkmi a nálepkou*). Postupné montovanie komponentov a medziproduktov končí vytvorením finálneho produktu – premyslenej vidlice.

Postupnosť krokov výrobného procesu, obmedzenia, informácie o dostupných zdrojoch, operáciách a ich vzájomná previazanosť, sú znalosti, ktoré musí aplikácia ukladať, spracovať a ponúknuť používateľovi vo forme aktuálne dostupných možností počas plánovania výrobného procesu.

Aplikácia zobrazuje vizualizovaný prechod výrobným procesom, kde používateľ vidí, ktoré aktivity je možné v danom okamihu vykonať (odlíšené modrou farbou – Obrázok 5). V ľavej časti okna aplikácie sa nachádza *Hodnotenie procesu*, ktoré umožňuje používateľovi:

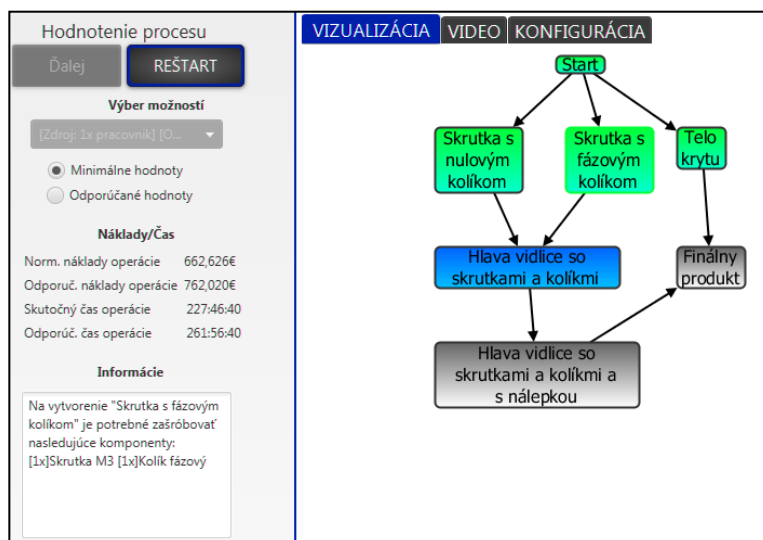
- meniť dostupné zdroje (počty pracovníkov a priemyselných robotov),
- časy spracovania nastaviť na požadované hodnoty,
- vidieť aktuálne vznikajúce náklady a čas operácie,
- vidieť doplňujúce informácie o montáži komponentov.





Obrázok 5 - Prechod výrobným procesom 1

Komponenty, ktoré prešli procesom montáže sú následné označené zelenou farbou tak ako to zobrazuje Obrázok 6. Pre lepšie pochopenie procesu výroby daného produktu umožňuje karta *Video* zobrazit' animovaný náhľad tohto procesu.



Obrázok 6 - Prechod výrobným procesom 2

Karta *Konfigurácia* umožňuje nastaviť výrobné kapacity, náklady a operácie, ktoré sa majú pri danom plánovaní výrobného procesu použiť.

VIZUALIZÁCIA	VIDEO	KONFIGURÁCIA		
<p><b>Cieľový produkt</b></p> <p>FinalProduct <input type="text"/></p> <p>Počet kusov <input type="text" value="100000"/></p> <p><b>Pracovná sila</b></p> <p>Počet ľudí <input type="text" value="3"/></p> <p>Mzda na pracovníka [€] <input type="text" value="400"/></p> <p><b>Roboty</b></p> <p>Počet robotov <input type="text" value="1"/></p> <p>Nákupná cena [€] <input type="text" value="100000"/></p> <p>Životnosť [hod] <input type="text" value="20000"/></p> <p>Spotreba energie [kWh] <input type="text" value="2"/></p>			<p><b>Všetky dostupné operácie</b></p> <ul style="list-style-type: none"> <li>Skrutkovanie kolíkov - operator</li> <li>Skrutkovanie kolíkov - robot</li> <li>Lepenie nalepky - operator</li> <li></li> <li></li> <li></li> <li></li> <li></li> </ul> <p>Obnoviť</p>	<p><b>Vybrané operácie</b></p> <div style="background-color: #ccccff; width: 100%; height: 100%;"></div> <p>Zmazať označené</p>

Obrázok 7 - Konfigurácia

Uvedená funkcionálna vytvára z aplikácie ľahko konfigurovateľnú a rýchlo dostupnú podporu plánovania výrobných procesov. Netreba však zabúdať na to, že kvalita aplikácie sa v značnej miere odvíja od kvality znalostí, ktorými aplikácia disponuje. Preto pri tvorbe znalostných systémov netreba zabúdať na doménových expertov, ktorí potrebnými znalosťami disponujú a znalostnými inžiniermi, ktorí tieto znalosti dokážu vložiť do systému.

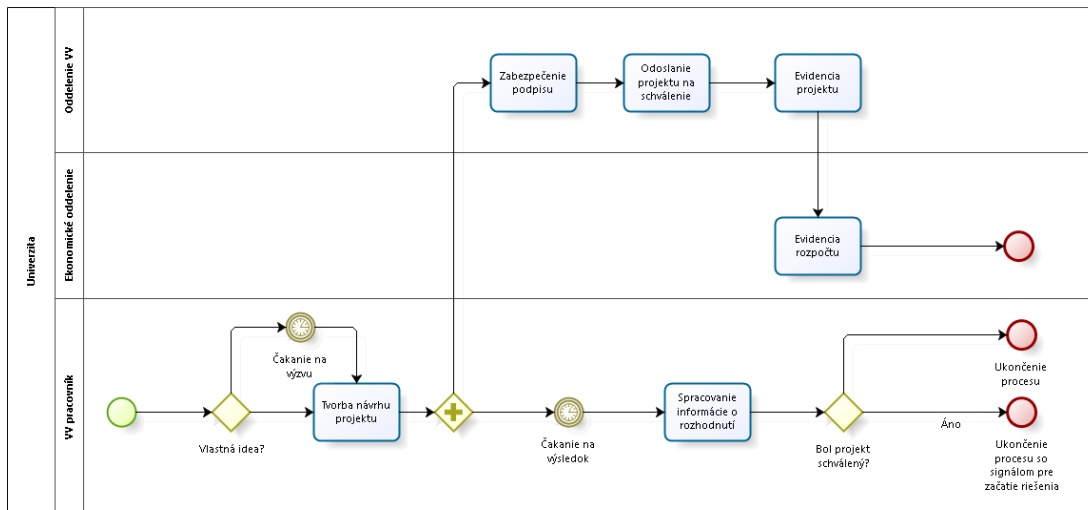
## 4.4 Použitie ontológie ako modelovacieho nástroja v MDA

Použitie ontológie ako modelovacieho nástroja v MDA môže odstrániť transformácie medzi rovinami CIM a PIM, ktoré sú potrebné pri použití grafických modelovacích jazykoch. Zjednotenie rovín CIM a PIM vyžaduje zjednotenie aj na úrovni notácie. Preto je dôležité v rámci jednotného ontologického zápisu (modelu) vybrať také prvky, ktoré by dokázali reprezentovať a zachytiť relevantné informácie z oboch rovín.

### 4.4.1 Modelovanie CIM a PIM úrovne pomocou ontológie

Rovina CIM býva často vyjadrená formou BPMN diagramov. Naproti tomu rovina PIM býva vyjadrená v jazyku UML. Obidve notácie (BPMN aj UML) sú značne rozsiahle a ich komplexné zachytenie by bolo nad rámec tejto práce. Preto sa autor zameria na niektoré vybrané a často používané elementy a diagramy.

Ako dáta pre výber elementov a tvorbu modelov budú slúžiť vybrané časti z procesu riadenia vedy a výskumu na Žilinskej univerzite.



Obrázok 8 – BPMN proces vybraných častí podania a schválenia vedecko-výskumného projektu na Žilinskej univerzite - spracované podľa projektu [7]

Vybrané BPMN elementy: *Plavecká dráha, Bazén, Aktivita, Začiatok procesu, Koniec procesu, Rozhodovací blok, Paralelné vetvenie, Časovač.*

Vybrané elementy UML diagramu prípadov použitia: *Aktér, Prípadovej použitia, Asociácia.*

Vzhľadom na uvedené elementy by model v rámci ontológie mal pozostávať z elementov, ktoré budú obsahovať nasledujúce prvky:

- ID elementu
- Názov
- Typ elementu
  - Začiatok (Start),
  - Koniec (End),
  - Aktivita (Activity),
  - Časovač (Timer),
  - Paralelný blok (Parallel),
  - Rozhodovací blok (Decision)
- ID Zdroja
- ID Cieľa
- Vykonávateľ
- Vlastník

#### **4.4.2 Dáta pre ontologický model**

Vzorové dáta pre naplnenie ontologického modelu sú vytvorené priamo z informácií pochádzajúcich z procesov riadenia vedy a výskumu na Žilinskej univerzite. Tieto procesy boli navrhnuté v rámci projektu *Rozvoj ľudských zdrojov s podporou integrovaného informačného systému na hodnotenie vedecko-výskumných výsledkov* [7]. Navrhnuté dáta zobrazuje Tabuľka 1.

Tabuľka 1 – Návrh dát pre ontologický model

<b>ID elementu:</b>	1	<b>ID elementu:</b>	6	<b>ID elementu:</b>	11
<b>Názov:</b>	-	<b>Názov:</b>	Čakanie na výsledok	<b>Názov:</b>	Zabezpečenie podpisu
<b>Typ:</b>	Start	<b>Typ:</b>	Timer	<b>Typ:</b>	Activity
<b>ID zdroja:</b>	-	<b>ID zdroja:</b>	5	<b>ID zdroja:</b>	5
<b>ID cieľa:</b>	2	<b>ID cieľa:</b>	7	<b>ID cieľa:</b>	12
<b>Vykonávateľ:</b>	VV pracovník	<b>Vykonávateľ:</b>	VV pracovník	<b>Vykonávateľ:</b>	Oddelenie VV
<b>Vlastník:</b>	Univerzita	<b>Vlastník:</b>	Univerzita	<b>Vlastník:</b>	Univerzita
<b>ID elementu:</b>	2	<b>ID elementu:</b>	7	<b>ID elementu:</b>	12
<b>Názov:</b>	Vlastná idea?	<b>Názov:</b>	Spracovanie informácie o rozhodnutí	<b>Názov:</b>	Odoslanie projektu na schválenie
<b>Typ:</b>	Decision	<b>Typ:</b>	Activity	<b>Typ:</b>	Activity
<b>ID zdroja:</b>	1	<b>ID zdroja:</b>	6	<b>ID zdroja:</b>	11
<b>ID cieľa:</b>	3, 4	<b>ID cieľa:</b>	8	<b>ID cieľa:</b>	13
<b>Vykonávateľ:</b>	VV pracovník	<b>Vykonávateľ:</b>	VV pracovník	<b>Vykonávateľ:</b>	Oddelenie VV
<b>Vlastník:</b>	Univerzita	<b>Vlastník:</b>	Univerzita	<b>Vlastník:</b>	Univerzita
<b>ID elementu:</b>	3	<b>ID elementu:</b>	8	<b>ID elementu:</b>	13
<b>Názov:</b>	Čakanie na výzvu	<b>Názov:</b>	Bol projekt schválený	<b>Názov:</b>	Evidencia projektu
<b>Typ:</b>	Timer	<b>Typ:</b>	Decision	<b>Typ:</b>	Activity
<b>ID zdroja:</b>	2	<b>ID zdroja:</b>	7	<b>ID zdroja:</b>	12
<b>ID cieľa:</b>	4	<b>ID cieľa:</b>	9, 10	<b>ID cieľa:</b>	14
<b>Vykonávateľ:</b>	VV pracovník	<b>Vykonávateľ:</b>	VV pracovník	<b>Vykonávateľ:</b>	Oddelenie VV
<b>Vlastník:</b>	Univerzita	<b>Vlastník:</b>	Univerzita	<b>Vlastník:</b>	Univerzita
<b>ID elementu:</b>	4	<b>ID elementu:</b>	9	<b>ID elementu:</b>	14
<b>Názov:</b>	Tvorba návrhu projektu	<b>Názov:</b>	Ukončenie procesu	<b>Názov:</b>	Evidencia rozpočtu
<b>Typ:</b>	Activity	<b>Typ:</b>	End	<b>Typ:</b>	Activity
<b>ID zdroja:</b>	2, 3	<b>ID zdroja:</b>	8	<b>ID zdroja:</b>	13
<b>ID cieľa:</b>	5	<b>ID cieľa:</b>	-	<b>ID cieľa:</b>	15
<b>Vykonávateľ:</b>	VV pracovník	<b>Vykonávateľ:</b>	VV pracovník	<b>Vykonávateľ:</b>	Ekonomické oddelenie
<b>Vlastník:</b>	Univerzita	<b>Vlastník:</b>	Univerzita	<b>Vlastník:</b>	Univerzita
<b>ID elementu:</b>	5	<b>ID elementu:</b>	10	<b>ID elementu:</b>	15
<b>Názov:</b>	-	<b>Názov:</b>	Ukončenie procesu so signálom pre začatie riešenia	<b>Názov:</b>	-
<b>Typ:</b>	Parallel	<b>Typ:</b>	End(Y)	<b>Typ:</b>	End
<b>ID zdroja:</b>	4	<b>ID zdroja:</b>	8	<b>ID zdroja:</b>	14
<b>ID cieľa:</b>	6, 11	<b>ID cieľa:</b>	-	<b>ID cieľa:</b>	-
<b>Vykonávateľ:</b>	VV pracovník	<b>Vykonávateľ:</b>	VV pracovník	<b>Vykonávateľ:</b>	Ekonomické oddelenie
<b>Vlastník:</b>	Univerzita	<b>Vlastník:</b>	Univerzita	<b>Vlastník:</b>	Univerzita

#### 4.4.3 Ontologický model

Na základe návrhu dát (Tabuľka 1) bol následne v nástroji Protege 4.3 (ontologický editor) vytvorený konkrétny ontologický model zapísaný v jazyku OWL. Keďže vytvorený OWL súbor je príliš rozsiahly na to, aby sa dal prehľadne zobraziť v tejto práci, na ukážku bola vybraná časť zápisu, ktorá sa venuje spomínanému elementu s ID 4. Relevantné časti sú zvýraznené žltou farbou.

```
<!-- http://.../ontologies/2016/3/UniverzitaOntology#4 -->
<owl:NamedIndividual rdf:about="http://.../ontologies/2016/3/UniverzitaOntology#4">
  <rdf:type rdf:resource="http://.../ontologies/2016/3/UniverzitaOntology#Element"
/>
  <rdf:type>
    <owl:Restriction>
      <owl:onProperty
rdf:resource="http://.../ontologies/2016/3/UniverzitaOntology#maTyp" />
      <owl:hasValue
rdf:resource="http://.../ontologies/2016/3/UniverzitaOntology#Activity" />
    </owl:Restriction>
  </rdf:type>
  <rdf:type>
    <owl:Restriction>
      <owl:onProperty
rdf:resource="http://.../ontologies/2016/3/UniverzitaOntology#maVlastnika" />
      <owl:hasValue
rdf:resource="http://.../ontologies/2016/3/UniverzitaOntology#Univerzita" />
    </owl:Restriction>
  </rdf:type>
  <rdf:type>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Restriction>
          <owl:onProperty
rdf:resource="http://.../ontologies/2016/3/UniverzitaOntology#maZdroj" />
          <owl:hasValue
rdf:resource="http://.../ontologies/2016/3/UniverzitaOntology#2" />
        </owl:Restriction>
        <owl:Restriction>
          <owl:onProperty
rdf:resource="http://.../ontologies/2016/3/UniverzitaOntology#maZdroj" />
          <owl:hasValue
rdf:resource="http://.../ontologies/2016/3/UniverzitaOntology#3" />
        </owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
  </rdf:type>
  <rdf:type>
    <owl:Restriction>
      <owl:onProperty
rdf:resource="http://.../ontologies/2016/3/UniverzitaOntology#maCiel" />
      <owl:hasValue
rdf:resource="http://.../ontologies/2016/3/UniverzitaOntology#5" />
    </owl:Restriction>
  </rdf:type>
  <rdf:type>
    <owl:Restriction>
      <owl:onProperty
rdf:resource="http://.../ontologies/2016/3/UniverzitaOntology#maVykonavateľa" />
      <owl:hasValue
rdf:resource="http://.../ontologies/2016/3/UniverzitaOntology#VV pracovnik" />
    </owl:Restriction>
  </rdf:type>
  <maMeno>Tvorba návrhu projektu</maMeno>
</owl:NamedIndividual>
```

#### 4.4.4 Mapovanie ontológie na úroveň CIM

Úroveň CIM býva reprezentovaná biznis modelmi. Aby mohla ontológia slúžiť pre potreby modelovania úrovne CIM je potrebné z ontologického modelu, ktorý je reprezentovaný súborom vo formáte OWL vytvoriť biznis model.

Na tvorbu týchto modelov je využívaná notácia BPMN (štandardizovaná skupinou OMG), ktorá poskytuje univerzálnu grafickú notáciu pre modelovanie biznis procesov. Na ukládanie BPMN modelov sa využíva štandard XPDŁ (štandardizovaný skupinou WfMC), ktorý umožňuje uložiť celú definíciu BPMN procesu formou špecifických XML súborov. XPDŁ predstavuje všeobecne akceptovaný formát, ktorý je využívaný v nástrojoch pre modelovanie biznis procesov.

Na vytvorenie biznis modelu z ontologického modelu bola vytvorená aplikácia, ktorá z OWL súboru automaticky vygeneruje XPDŁ dokument popisujúci biznis proces.

Na to, aby bolo možné mapovať ontologický model na biznis procesy s elementami špecifikovanými v podkapitole 4.4.1 - *Plavecká dráha, Bazén, Aktivita, Začiatok procesu, Koniec procesu, Rozhodovací blok, Paralelné vetvenie a Časovač* je potrebné definovať mapovacie pravidlá a štruktúru pre XPDŁ súbor.

Mapovanie využíva *Vlastnosti objektov* (Object properties) definované v ontologickom modeli:

1. Plavecká dráha sa mapuje cez *maVykonavateľa*
2. Bazén sa mapuje cez *maVlastníka*
3. Mapovanie typov elementov prebieha cez vlastnosť *maTyp*
  - Aktivita - hodnota *Activity*
  - Začiatok - hodnota *Start*
  - Koniec - hodnota *End*
  - Rozhodovací blok - hodnota *Decision*
  - Paralelné vetvenie - hodnota *Parallel*
  - Časovač - hodnota *Timer*
4. Názvy elementov sú mapované cez vlastnosť *maMeno*
5. Mapovanie prepojení elementov je riešené cez vlastnosti *maZdroj* a *maCiel*

Štruktúra XPDL musí v tomto prípade obsahovať najmä elementy ako: *Pools, Pool, Lanes, Lane, WorkflowProcesses, WorkflowProcess, Activity, Event, StartEvent, EndEvent, IntermediateEvent, Condition, Route, Transition*. Tento XPDL súbor je možné otvoriť priamo v modelovacom nástroji Bizagi Modeler a tak zobrazit' biznis proces v plne grafickej podobe.

#### 4.4.5 Mapovanie ontológie na úroveň PIM

Jedným z modelov, ktorý sa používa na reprezentovanie úrovne platformovo nezávislého modelu – PIM je aj diagram prípadov použitia. Diagramy vyjadrené v modelovacom jazyku UML (medzi ktoré patrí aj diagram prípadov použitia) bývajú ukladané vo formáte XMI – jedná sa o všeobecne uznávaný štandard vydaný skupinou OMG.

Na to, aby bolo možné mapovať ontologický model do diagramu prípadov použitia s elementami špecifikovanými v podkapitole 4.4.1 - *Aktér, Prípád použitia, Asociácia* je potrebné definovať mapovacie pravidlá a štruktúru pre XMI súbor.

Mapovanie využíva *Vlastnosti objektov* (Object properties) definované v ontologickom modeli:

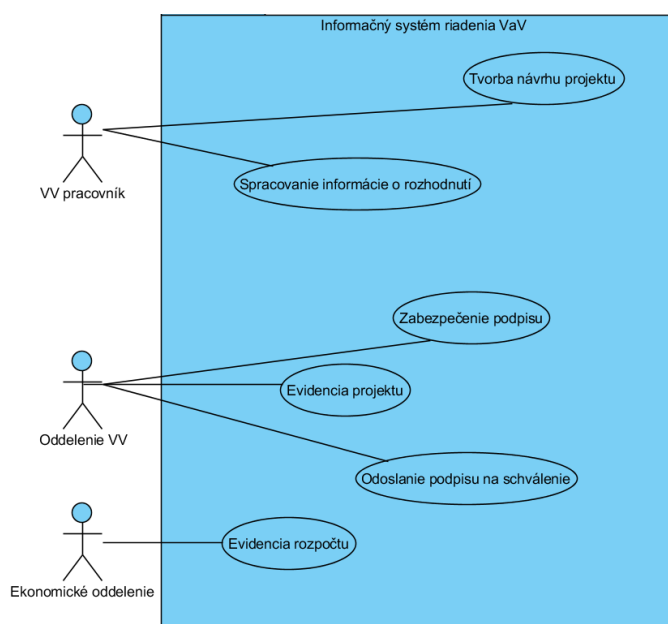
1. Aktéri sa mapujú cez *maVykonavateľa*
2. Prípady použitia sa mapujú cez *maTyp* – hodnota *Activity*
3. Mapovanie asociácii vychádza z predchádzajúcich dvoch bodov, pretože každá aktivita má priradeného vykonávateľa - tým pádom je možné jednoznačne určiť asociáciu.

Štruktúra XMI súborov dokáže byť pomerne zložitá a môže obsahovať veľké množstvo elementov a atribútov. Hľadanie vhodnej XMI štruktúry prebiehalo na základe experimentov. Okrem všeobecných atribútov (názov, verzia, typ diagramu, ...) je pre zapísanie elementov *Aktér, Prípád použitia, Asociácia* potrebné použiť atribúty *uml:Actor, uml:UseCase, uml:Association* a *uml:Model* pre vytvorenie hraníc systému.

Na základe uvedených poznatkov bola vytvorená aplikácia, ktorá dokáže mapovať vytvorený ontologický model na XMI súbor.

Následne je možné tento súbor otvoriť v modelovacom nástroji, ktorý podporuje UML notáciu a umožňuje import vo formáte XMI. Grafickú formu diagramu prípadov použitia po importovaní XMI súboru do nástroja Visual Paradigm znázorňuje Obrázok 9 uvedený nižšie.





Obrázok 9 - Podanie a schválenie projektu - diagram prípadov použitia

#### 4.4.6 Zhrnutie

Prezentované riešenie mapovania ontológie na úroveň CIM a PIM ukázalo, ako je možné využiť ontológiu pre jednotné modelovanie spomínaných úrovní v rámci MDA.

Ontologický model je možné mapovať na *Model nezávislý na počítačovom spracovaní* (CIM) a aj na *Model nezávislý na platforme* (PIM). Následne je možné tieto modely zobrazit' v plnohodnotnej grafickej forme. Týmto spôsobom je možné zrýchliť modelovací proces a takisto odstrániť potrebu transformácie medzi úrovňami CIM a PIM. Na to, aby bolo možné odstrániť potrebu transformácie v plnom rozsahu by bolo nutné dané mapovanie rozšíriť na celé spektrum problematiky vytvárania modelov v úrovniach CIM a PIM s ohľadom na všetky notácie, obmedzenia, konvencie, typy súborov a štandardy. Napriek tomu súčasné riešenie dokáže zjednotiť, zrýchliť a sprehl'adniť návrh, tvorbu a údržbu biznis procesov a diagramov prípadov použitia z hľadiska CIM/PIM transformácii a poskytuje priestor pre ďalší rozvoj v tejto výskumnej oblasti.

Výhodou navrhnutého riešenia je aj to, že prípadné zmeny v špecifikácii počas návrhu IS stačí meniť iba v ontologickom modeli a následne sa zmeny potom cez mapovanie na CIM/PIM úroveň prejavia priamo vo výsledných modeloch. Vďaka tomuto faktoru nie je potrebné prácne manuálne upravovať všetky vytvorené modely pri každej zmene.

## 5 Zhodnotenie riešenia

Potreba pracovať súčasne s informáciami, dátami aj znalosťami vytvára nové výzvy, vyriešenie ktorých môže posúvať vývoj informačných systémov na vyššiu úroveň.

Použitie ontológií pri vývoji IS je úzko späté s ontologickým a znalostným inžinierstvom. Ontologické inžinierstvo sa v niektorých ohľadoch veľmi nelíši od procesu vývoja informačných systémov, ktorý je charakteristický analýzou, tvorbou systémových modelov, a tvorbou logického a fyzického návrhu systému. Návrh previazania modelom riadenej architektúry s ontológiu je prvým prínosom.

S neustále pribúdajúcimi dátami, informáciami a znalosťami sa dopyt po znalostných systémoch a manažmente znalostí zvyšuje. Napriek tomu nebola všeobecnému postupu tvorby znalostných systémov doteraz venovaná väčšia pozornosť. Vytvorenie architektonického rámca podľa princípov normy ISO/IEC/IEEE 42010:2011 *Systems and Software Engineering* pre vývoj znalostných systémov je druhým prínosom. Následné aplikovanie vytvoreného architektonického rámca pre špecifickú doménu s experimentálnym overením analýzy a návrhu IS je tretím prínosom.

Štvrtým prínosom tejto dizertačnej práce je pozitívne overenie predpokladu, že ontológia by mohla byť použitá ako modelovací jazyk pre roviny CIM (Computer Independent Model) a PIM (Platform Independent Model) v modelom riadenej architektúre.

Uvedené prínosy otvárajú nové možnosti pre vývoj informačných systémov z hľadiska ich všeobecného návrhu (s dôrazom na znalostné systémy) a z hľadiska modelom riadenej architektúry a transformácii rovín CIM a PIM.

## Záver

Cieľom tejto dizertačnej práce bolo navrhnuť architektonický rámec riešenia informačného systému pre riadenie znalostí v organizácii použitím princípov modelom riadenej architektúry (MDA) a ontológií ako modelovacieho jazyka v rovinách CIM (Computer Independent Model) a PIM (Platform Independent Model). Po začlenení princípov modelom riadenej architektúry do znalostného inžinierstva bol vytvorený architektonický rámec pre vývoj znalostných systémov. Výsledným riešením experimentálneho overenia je systém pre podporu plánovania procesov priemyselnej výroby. V rámci 4. parciálneho cieľa bola navrhnutá dátová štruktúra a vytvorený ontologický model. Ontologický model, ktorého účelom bolo navzájom integrovať informácie z rovin CIM a PIM bol vytvorený v jazyku OWL. Po definovaní transformačných pravidiel a súborových štruktúr bol model mapovaný na biznis procesy v jazyku BPMN a prípady použitia v jazyku UML. Takto vytvorené modely je možné načítať v modelovacích nástrojoch a zobrazit' ich priamo v grafickej podobe. Toto riešenie podporuje vyslovený predpoklad – *Ontológia by mohla byť vhodnejším nástrojom pre uplatnenie princípov v MDA, pretože nebude potrebné vytvárať transformačné vzťahy medzi CIM a PIM rovinou*. Bežne používané modelovacie nástroje síce umožňujú tvorbu modelov do najmenších detailov, avšak neriešia problematiku priamej transformácie medzi vytvorenými modelmi.

Použitie ontologického modelu oproti iným spôsobom uloženia je odôvodnené tým, že ontológia nie je iba úložiskom ale poskytuje mechanizmy pre reprezentáciu dát, informácii a znalostí, ich vzájomné prepájanie a vyvodzovanie súvislostí. Vzhľadom na tento fakt možno v budúcnosti vytvorený ontologický model efektívne rozšíriť nielen o podporu iných modelovacích jazykov a notácii ale aj o aspekty, ktorých vzájomná integrácia v rámci návrhu a modelovania IS nebola doteraz uvažovaná. Môže to byť priama integrácia na projektový manažment, dokumentácie, existujúce dáta či dokonca celé informačné systémy. Toto je však iba niekoľko námetov na ďalšie možnosti smerovania v tejto oblasti.

Takisto netreba zabúdať to, že samotná tvorba komplexných ontologických modelov vyžaduje rozsiahle skúsenosti znalostného inžiniera.

## Použitá literatúra a zdroje

- [1] Good e-Learning Blog, „Common BPMN modeling mistakes: Swimlanes,“ [Online]. Available: <http://blog.godelearning.com/bpmn/common-bpmn-modeling-mistakes-swimlanes/>. [Cit. 7 Október 2014].
- [2] I. 42010:2011, *Systems and software engineering - Architecture description*, 2011.
- [3] Roger Sessions ObjectWatch, Inc., „A Comparison of the Top Four Enterprise-Architecture Methodologies: Microsoft developer network,“ Máj 2007. [Online]. Available: <https://msdn.microsoft.com/en-us/library/bb466232.aspx>. [Cit. 4 Apríl 2015].
- [4] N. Rozanski a E. Woods, *Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives*, New Jersey: Pearson Education, Inc., 2005.
- [5] L. Gála, A. Buchalcevová a J. Jandoš, „Podniková architektura,“ 2012.
- [6] „Elektroinštalačný materiál - Toms Hardware,“ 2016. [Online]. Available: <http://www.tomshardware.sk/p/11262/vidlica-380v-sez-ivn-3253-32a-ip44-na-kabel>.
- [7] Žilinská univerzita, ITMS 26110230063, *Rozvoj ľudských zdrojov s podporou integrovaného informačného systému na hodnotenie vedecko-výskumných výsledkov*.

## Zoznam vlastných publikovaných prác

- [1] Ontologies and knowledge base / Boris Bučko, Vladimír Smataník. In: International journal of engineering sciences & research technology (IJESRT) [elektronický zdroj]. - ISSN 2277-9655. - Vol. 3 no. 10 (2014), online, s. 490-493.
- [2] Reasons for investigating of alternative solutions of is development / Boris Bučko, Matilda Drozdová. In: International journal of information technology and business management [elektronický zdroj]. - ISSN 2304-0777. - Vol. 27 (29 June 2014), online, s. 79-84. - Popis urobený 16.01.2015.
- [3] Using ontologies in information systems development / Boris Bučko, Matilda Drozdová. In: Časopis Slovenskej spoločnosti pre systémovú integráciu [elektronický zdroj]. - ISSN 1336-5916. - No. 1 (2014), online, [4] s. - Popis urobený 15.4.2015.
- [4] Ontology as a base for state space of expert systems / Bučko Boris ... [et al.]. In: Web Intelligence (WI) and Intelligent Agent Technologies (IAT) [elektronický zdroj] : IEEE/WIC/ACM international joint conferences : 11-14 August 2014, Warsaw, Poland : proceedings. - [S.l.]: IEEE, 2014. - ISBN 978-1-4799-4143-8. - USB, s. 408-413.
- [5] Transformation in model driven architecture / Bučko Boris ... [et al.]. In: Information systems architecture and technology : proceedings of 36th international conference on Information systems architecture and technology - ISAT 2015. - [S.l.]: Springer, 2016. - ISBN 978-3-319-28553-5. - S. 193-203. - (Advances in intelligent systems and computing, Vol. 429. - ISSN 2194-5357).
- [6] Inferring over functionally equivalent ontologies / Boris Bučko ... [et al.]. In: ICDICP2015 : the fifth international conference on Digital information processing and communications : October 7-9, 2015, Sierre, Switzerland : proceedings. - [S.l.]: IEEE, 2015. - ISBN 978-1-4673-6831-5. - S. 201-206.
- [7] Building ontologies from PPR hubs / Boris Bučko ... [et al.]. In: INES 2015 [elektronický zdroj] : 19th international conference on Intelligent engineering systems : proceedings : September 3-5, 2015 Bratislava, Slovakia. - [S.l.]: IEEE, 2015. - ISBN 978-1-4673-7938-0. - USB kľúč, s. 165-168.
- [8] Architectures of the next eLearning systems upgrade / M. Drozdová, B. Bucko, and I. Brídová. In: ICETA 2014 : 12th IEEE international conference on Emerging eLearning technologies and applications : December 4-5, 2014, Stary Smokovec, Slovakia. - [S.l.]: IEEE, 2014. - ISBN 978-1-4799-7739-0. - S. 109-114.
- [9] Knowledge in information systems / Boris Bučko, Katarína Zábovská. In: ScienFIST.org : international journal of information technologies, engineering and management science. - ISSN 1339-9470. - Vol. 1, iss. 1 (2015), online, s. 4-6.

## Ohlasy

- [1] **Knowledge in information systems / Boris Bučko, Katarína Zábovská. In: ScienFIST.org : international journal of information technologies, engineering and management science. - ISSN 1339 -9470. - Vol. 1, iss. 1 (2015), online, s. 4-6.**
- [1] **2015 JANČUŠOVÁ, M., ĎURICA, J. Prototyp meracieho zariadenia na báze bezobslužných technológií. In: ProIN. ISSN 1339-2271, 2015, roč. 16, č. č, s. 29-32.**
- [2] **2015 JANČUŠOVÁ, M., GRZNÁR, P. Manufacturing and logistics strategy: product design and development decisions related to assembly. In: International Journal of Mechanical Engineering and Automation. ISSN 2333-9179, 2015, vol. 2, iss. 11, s. 505-509.**
- [3] **2015 LNĚNIČKA, M. Reframing architecture frameworks using cloud computing and open (big) data. In: ScienFIST.org. ISSN 1339--9470, 2015, vol. 1, iss. 2, s. 1-5.**